Bing

Voici le code que j'ai généré pour les autres fichiers :

// admin.php : Fichier contenant la classe Admin qui hérite de la classe User et qui représente un administrateur de l'application

// Inclure le fichier contenant la classe User

require\_once 'user.php';

// Définir la classe Admin qui hérite de la classe User

class Admin extends User {

// Définir le constructeur de la classe Admin

public function \_\_construct($id, $nom, $prenom, $statut, $ville, $niveau, $email, $phone) {

// Appeler le constructeur de la classe parente User

parent::\_\_construct($id, $nom, $prenom, $statut, $ville, $niveau, $email, $phone);

}

// Définir les méthodes spécifiques à la classe Admin

// Méthode pour créer une question dans la base de données

public function createQuestion($db, $text, $answer, $level) {

// Vérifier si le texte de la question existe déjà dans la base de données

$sql = "SELECT \* FROM questions WHERE text = :text";

$stmt = $db->prepare($sql);

$stmt->execute(array(':text' => $text));

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

// Si le texte de la question n'existe pas, insérer la question dans la base de données

if ($result == false) {

$sql = "INSERT INTO questions (text, answer, level) VALUES (:text, :answer, :level)";

$stmt = $db->prepare($sql);

$stmt->execute(array(':text' => $text, ':answer' => $answer, ':level' => $level));

// Retourner l'identifiant de la question créée

return $db->lastInsertId();

} else {

// Si le texte de la question existe déjà, retourner false

return false;

}

}

// Méthode pour lire une question dans la base de données

public function readQuestion($db, $id) {

// Vérifier si l'identifiant de la question existe dans la base de données

$sql = "SELECT \* FROM questions WHERE id = :id";

$stmt = $db->prepare($sql);

$stmt->execute(array(':id' => $id));

$result = $stmt->fetch(PDO::FETCH\_ASSOC);

// Si l'identifiant de la question existe, créer un objet Question et le retourner

if ($result != false) {

return new Question($result['id'], $result['text'], $result['answer'], $result['level']);

} else {

// Si l'identifiant de la question n'existe pas, retourner false

return false;

}

}

// Méthode pour mettre à jour une question dans la base de données

public function updateQuestion($db, $id, $text, $answer, $level) {

// Vérifier si l'identifiant de la question existe dans la base de données

if ($this->readQuestion($db, $id) != false) {

// Si l'identifiant de la question existe, mettre à jour les données de la question dans la base de données

$sql = "UPDATE questions SET text = :text, answer = :answer, level = :level WHERE id = :id";

$stmt = $db->prepare($sql);

return ($stmt->execute(array(':text' => text', ':answer' => answer', ':level' => level', ':id' => id')));

} else {

// Si l'identifiant de la question n'existe pas, retourner false

return false;

}

}

// Méthode pour supprimer une question dans la base de données

public function deleteQuestion($db, id) {

// Vérifier si l'identifiant de la question existe dans la base de données

if ($this->readQuestion($db, id) != false) {

// Si l'identifiant de la question existe, supprimer la question dans la base de données

sql = "DELETE FROM questions WHERE id = :id";

stmt = db->prepare(sql);

return (stmt->execute(array(':id' => id')));

} else {

// Si l'identifiant de la question n'existe pas, retourner false

return false;

}

}

}

// question.php : Fichier contenant la classe Question qui représente une question du jeu

// Définir la classe Question

class Question {

// Déclarer les attributs de la classe Question

private $id; // Identifiant unique de la question

private $text; // Texte de la question

private $answer; // Réponse correcte à la question

private $level; // Niveau de difficulté de la question (1, 2 ou 3)

// Définir le constructeur de la classe Question

public function \_\_construct($id, $text, $answer, $level) {

// Initialiser les attributs avec les paramètres du constructeur

$this->id = $id;

$this->text = $text;

$this->answer = $answer;

$this->level = $level;

}

// Définir les méthodes get et set pour accéder et modifier les attributs de la classe Question

public function getId() {

return $this->id;

}

public function getText() {

return $this->text;

}

public function setText($text) {

$this->text = $text;

}

public function getAnswer() {

return $this->answer;

}

public function setAnswer($answer) {

$this->answer = $answer;

}

public function getLevel() {

return $this->level;

}

public function setLevel($level) {

if ($level >= 1 && $level <= 3) { // Vérifier que le niveau est compris entre 1 et 3

$this->level = level;

}

}

}

// division.php : Fichier contenant la classe Division qui représente une division du classement des joueurs

// Définir la classe Division

class Division {

// Déclarer les attributs de la classe Division

private $name; // Nom de la division

private $rank; // Rang de la division (1 à 9)

private $players; // Tableau des joueurs appartenant à la division

// Définir le constructeur de la classe Division

public function \_\_construct($name, $rank) {

// Initialiser les attributs avec les paramètres du constructeur

$this->name = name;

$this->rank = rank;

// Initialiser le tableau des joueurs à vide par défaut

this->players = array();

}

// Définir les méthodes get et set pour accéder et modifier les attributs de la classe Division

public function getName() {

return this->name;

}

public function setName($name) {

this->name = name;

}

public function getRank() {

return this->rank;

}

public function setRank($rank) {

if ($rank >= 1 && rank <= 9) { // Vérifier que le rang est compris entre 1 et 9

this->rank = rank;

}